Paper Session 1

Software Evolution

Paul Klint, Bas van Vlijmen, Jurgen Vinju

Today

- Writing your Motivation
  - Motivation
  - Paragraphs
  - Improving text and submitting it
- Discuss clones, metric & design patterns
- Intro to argumentation

Motivation

- What is the goal of a motivation?

  - Who are you motivating?

- What are you motivating?

  - giving excuses for invested time in writing? no.

  - providing reasons for them to read your paper!

- A typical motivation includes:

  - Definition, position, contribution, plan

Definition

- What is the topic?
- What is not the topic?

- What is a good definition?

Position

- What is your perspective? goal?

- In what sense is this paper "unique" or "special"?

- Position by relating your paper

  - to other papers ("related work")

  - to what existed before, to current reality

  - to what the reader already knows

  - to a known or accepted contrary view

Contribution

- What is the return on investment for reading?

- What are the lessons to learn?

- What are actionable conclusions?

- How {c,w,sh}ould your paper "change the world"?

- Who knows examples?

Plan

- Outline of the paper
  - forward references to sections
  - explains global structure of the paper
- Helps the reader decide where to put energy

- (not relevant for your current assignment)

Paragraphs

- Good paragraphs have

  - correct and simple <u>language</u> (ok)

  - helpful and clear <u>structure</u>

  - <u>logical</u> message

  - readable <u>flow</u>

Paragraphs

- Bad paragraphs have:
  - spelling errors and grammatical mistakes
  - no (clear) structure
  - reasoning fallacies or incomplete arguments
  - incoherent and unconnected sentences

(These and the following examples are adapted from

"Style - the basics of clarity and grace", by J.M. Williams)

Tip 1

- Do not confuse 'difficult to understand' with 'deep thinking'

- "An understanding of the causal factors involved in code cloning by programmers could lead to more effective prevention."

- "We could prevent programmers to clone code more effectively if we new why they do."

- What is the difference?

Tip 2

- Writing is telling a story
  - Make main characters <u>subjects</u>
  - Make important actions <u>verbs</u>
- "Once upon a time, as a <u>walk</u> in the woods was taking place on the part of <u>Little Red Riding Hood</u>, the Wolf's jump from behind a tree…"
- "Once upon a time, <u>Little Red Riding Hood</u> was <u>walking</u> in the forest, when the Wolf jumped

Clones & fairy tales

- "The <u>argument</u> of <u>authors</u> of code clone detection tools with respect to the harmfulness of copy and paste programming is based on their <u>belief</u> in the tendency of <u>programmers</u> of <u>wanting</u> to maintain each clone consistently."

- "The <u>authors</u> of code clone detection tools <u>argue</u> that copy and paste programming is harmful because they <u>believe</u> that <u>programmers</u> <u>want</u> to maintain each clone consistently."

- So: use subjects for actors and verbs for actions.

- This makes your paragraphs less abstract and more concrete

Tip 3

- From quality of sentences to paragraphs

- Connect sentences, sometimes making a sentence less clear to make the whole paragraph more clear

- "Programmers frequently introduce clones: exact copies of source code fragments. Later, they may forget to maintain such clones consistently"

- "Programmers frequently introduce clones: exact copies of source code fragments. Such clones are often forgotten to be maintained consistently"

- Flow is especially important in longer paragraphs

- Use the passive tense to flip a sentence

Tip 4: Get the book

- Style - The Basics of Clarity and Grace
  - Second edition
  - Joseph M. Williams
  - 2006
  - Pearson Education, Inc

Feedback time (15 min)

- Read each others motivation (on a different topic), and suggest improvements.
  - Do you understand the text?
  - Are you motivated to read on?
- Identify structure:
  - topic, explanation, illustration
  - abstract to concrete
  - bottom-line up front?
- Identify logic:
  - cause & effect
  - "if", "because", "when", "so"
- Identify flow
  - order of topics in consecutive sentences
  - connectives "so", "because of ...", "therefore", "this is why", "these examples show", ...

Discussion

- Did you recognize bad/good style?

- Was is doable to improve? How?

- What about the topics? Questions?

  - Clones

  - Metrics

  - Patterns

Rewrite your motivation

- 15 minutes
- submit it on blackboard after (today!)
- we check if it's ok

Next session

- Read about argumentation and schemes

- Choose a position

- Make argumentation scheme

- Less is more: make <u>one</u> interesting point

- <u>Try</u> to complete it, all the way from claim down to facts

- <u>Stop</u> at well-motivated assumptions and (published?) conjectures, when time is up.

Argumentation

- What is the goal of an argument?

- What are the means to get to this goal?

Argumentation

- Correct logic

- Facts

- Clearly stated assumptions

- KISS - keep it simple stupid

- (convincing rhetoric and writing style are for session 3)

Wicked Problems

- [wikipedia] ""Wicked problem" is a phrase originally used in social <u>planning</u> to describe a problem that is difficult or impossible to solve because of incomplete, contradictory, and changing requirements that are often difficult to recognize.

- The term 'wicked' is used, not in the sense of evil but rather its resistance to resolution.<u>[1]</u> Moreover, because of complex <u>interdependencies,</u> the effort to solve one aspect of a wicked problem may reveal or create other problems."

- Software Engineering is a wicked problem

- Every part of software engineering probably too

- This is why argumentation is particularly interesting in our field

Use Schemes

- To separate improving logic from improving language

- Box is a statement, claim or fact

- Edge is a causal relation

- "Araucaria" is a software tool you could use
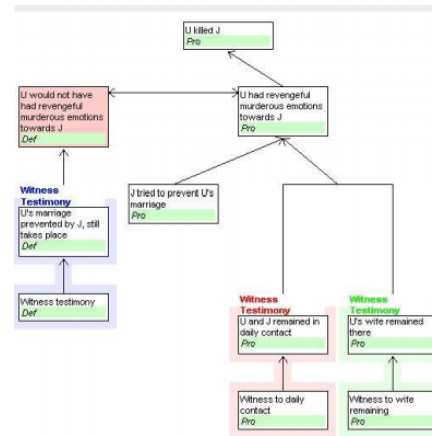
- Args are pro or con
- Optionally, you can distinguish conjunctions (and) from disjunctions (or)

Next session

- Read about argumentation and schemes
  - Van Eemeren (see BB)
- Make argumentation scheme